15. The method of claim **14**, further comprising determining the performance property and the security property based on information in a predefined database, the information including relationships between (i) software components usable in software stacks, (ii) performance characteristics of the software components, and (iii) security characteristics of the software components.

16. The method of claim **11**, wherein the predefined attribute involves the respective score for the particular software-stack candidate being a maximum score or a minimum score among a plurality of scores corresponding to the software-stack candidates.

17. The method of claim **11**, further comprising determining the software-stack candidates by performing the search using a heuristic search algorithm.

18. The method of claim **11**, further comprising determining the software-stack candidates by performing the search using a stochastic search algorithm.

19. The method of claim **11**, further comprising:

determining a recommended file to include in the recommended software-stack, wherein the recommended file is different from the target software item and is not a dependency of the target software item, and wherein the recommended file is configured to improve a performance characteristic or a security characteristic of the recommended software-stack; and

including the recommended file in the recommended software-stack.

20. A non-transitory computer-readable medium comprising program code that is executable by a processor for causing the processor to:

receive an input specifying a target software item and specifying a characteristic of a computing environment in which the target software item is to be executed;

generate software-stack candidates for the target software item by performing a search using a search algorithm configured to recursively analyze direct and indirect dependencies of the target software item, the software-stack candidates having unique configurations of software components including the target software item and dependencies of the target software item;

determine a respective score for each software-stack candidate of the software-stack candidates based on the characteristic of the computing environment and a unique configuration of software components forming the software-stack candidate;

select a particular software-stack candidate from the software-stack candidates as a recommended software-stack, based on the respective score for the particular software-stack candidate having a predefined attribute; and

cause the recommended software-stack to be included in the computing environment.

21. The system of claim **1**, wherein the target software item is an individual software application.

22. The system of claim **1**, wherein the software-stack candidates include at least 500 software-stack candidates, and wherein the search algorithm is an optimization algorithm configured to iteratively analyze the at least 500 software-stack candidates to determine an optimal software-stack candidate relative to the other software-stack candidates of the at least 500 software-stack candidates, the optimal software-stack candidate serving as the recommended software-stack candidate.

* * * * *